

本文介绍了 Python 操作 MYSQL、执行 SQL 语句、获取结果集、遍历结果集、取得某个字段、获取表字段名、将图片插入数据库、执行事务等各种代码实例和详细介绍，代码居多，是一桌丰盛唯美的代码大餐。

实例 1、取得 MYSQL 的版本

在 windows 环境下安装 mysql 模块用于 python 开发，请见我的另一篇文章：

[MySQL-python Windows 下 EXE 安装文件下载](#)

```
# -*- coding: UTF-8 -*-

#安装 MYSQL DB for python
import MySQLdb as mdb

con = None

try:
    #连接 mysql 的方法: connect('ip','user','password','dbname')
    con = mdb.connect('localhost', 'root',
        'root', 'test');

    #所有的查询，都在连接 con 的一个模块 cursor 上面运行的
    cur = con.cursor()

    #执行一个查询
    cur.execute("SELECT VERSION()")

    #取得上个查询的结果，是单个结果
    data = cur.fetchone()
    print "Database version : %s " % data
finally:
    if con:
        #无论如何，连接记得关闭
        con.close()
```

执行结果：

Database version : 5.5.25

实例 2、创建一个表并且插入数据

主要还是在 cursor 上面执行 execute 方法来进行，请见源码：

```
# -*- coding: UTF-8 -*-

# 来源于 www.crazyant.net 总结整理

import MySQLdb as mdb
import sys

#将 con 设定为全局连接
con = mdb.connect('localhost', 'root', 'root', 'test');

with con:

    #获取连接的 cursor，只有获取了 cursor，我们才能进行各种操作
    cur = con.cursor()
    #创建一个数据表 writers(id,name)
    cur.execute("CREATE TABLE IF NOT EXISTS \
        Writers(Id INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(25))")
    #以下插入了 5 条数据
    cur.execute("INSERT INTO Writers(Name) VALUES('Jack London')")
    cur.execute("INSERT INTO Writers(Name) VALUES('Honore de Balzac')")
    cur.execute("INSERT INTO Writers(Name) VALUES('Lion Feuchtwanger')")
    cur.execute("INSERT INTO Writers(Name) VALUES('Emile Zola')")
    cur.execute("INSERT INTO Writers(Name) VALUES('Truman Capote')")
```

运行结果（在 phpmyadmin 中查看，在 mysql 命令行查看结果是一样的）：


```
rows = cur.fetchall()

#依次遍历结果集，发现每个元素，就是表中的一条记录，用一个元组来显示
for row in rows:
    print row
```

运行结果：

(1L, 'Jack London')
(2L, 'Honore de Balzac')
(3L, 'Lion Feuchtwanger')
(4L, 'Emile Zola')
(5L, 'Truman Capote')

上面的代码，用来将所有的结果取出，不过打印的时候是每行一个元祖打印，现在我们使用
方法，取出其中的单个数据：

```
# -*- coding: UTF-8 -*-

# 来源：疯狂的蚂蚁的博客 www.crazyant.net 总结整理

import MySQLdb as mdb
import sys

#获取 mysql 的连接对象
con = mdb.connect('localhost', 'root', 'root', 'test');

with con:
    #获取执行查询的对象
    cur = con.cursor()

    #执行那个查询，这里用的是 select 语句
    cur.execute("SELECT * FROM Writers")

    #使用 cur.rowcount 获取结果集的条数
    numrows = int(cur.rowcount)

    #循环 numrows 次，每次取出一行数据
    for i in range(numrows):
        #每次取出一行，放到 row 中，这是一个元组(id,name)
        row = cur.fetchone()
        #直接输出两个元素
        print row[0], row[1]
```

运行结果：

1 Jack London

2 Honore de Balzac

3 Lion Feuchtwanger

4 Emile Zola

5 Truman Capote

- numrows = int(cur.rowcount)用于获取结果集的数目
- row = cur.fetchone()每次取出一行数据，同时记录集的指针执行下一行

实例 4、使用字典 **cursor** 取得结果集（可以使用表字段名字访问值）

```
# -*- coding: UTF-8 -*-

# 来源：疯狂的蚂蚁的博客 www.crazyant.net 总结整理

import MySQLdb as mdb
import sys

#获得 mysql 查询的连接对象
con = mdb.connect('localhost', 'root', 'root', 'test')

with con:
    #获取连接上的字典 cursor，注意获取的方法，
    #每一个 cursor 其实都是 cursor 的子类
    cur = con.cursor(mdb.cursors.DictCursor)

    #执行语句不变
    cur.execute("SELECT * FROM Writers")

    #获取数据方法不变
    rows = cur.fetchall()

    #遍历数据也不变（比上一个更直接一点）
    for row in rows:
        #这里，可以使用键值对的方法，由键名字来获取数据
        print "%s %s" % (row["Id"], row["Name"])
```

实例 5、获取单个表的字段名和信息的方法

```
# -*- coding: UTF-8 -*-

# 来源：疯狂的蚂蚁的博客 www.crazyant.net 总结整理

import MySQLdb as mdb
import sys

#获取数据库的连接对象
con = mdb.connect('localhost', 'root', 'root', 'test')

with con:
    #获取普通的查询 cursor
    cur = con.cursor()
    cur.execute("SELECT * FROM Writers")

    rows = cur.fetchall()

    #获取连接对象的描述信息
    desc = cur.description
    print 'cur.description:', desc

    #打印表头，就是字段名字
    print "%s %3s" % (desc[0][0], desc[1][0])

    for row in rows:
        #打印结果
        print "%2s %3s" % row
```

运行结果：

```
cur.description: (('Id', 3, 1, 11, 11, 0, 0), ('Name', 253, 17, 25, 25, 0, 1))
```

Id Name

1 Jack London

2 Honore de Balzac

3 Lion Feuchtwanger

4 Emile Zola

5 Truman Capote

实例 6、使用 Prepared statements 执行查询（更安全方便）

```
# -*- coding: UTF-8 -*-
```

```
# 来源：疯狂的蚂蚁的博客 www.crazyant.net 总结整理

import MySQLdb as mdb
import sys

con = mdb.connect('localhost', 'root', 'root', 'test')

with con:

    cur = con.cursor()
    #我们看到，这里可以通过写一个可以组装的 sql 语句来进行
    cur.execute("UPDATE Writers SET Name = %s WHERE Id = %s",
                ("Guy de Maupasant", "4"))
    #使用 cur.rowcount 获取影响了多少行
    print "Number of rows updated: %d" % cur.rowcount
```

结果：

Number of rows updated: 1

实例 7、把图片用二进制存入 MYSQL

有人喜欢把图片存入 **MYSQL**（这种做法貌似很少吧），我看大部分的程序，图片都是存放在服务器上的文件，数据库中存的只是图片的地址而已，不过 **MYSQL** 是支持把图片存入数据库的，也相应的有一个专门的字段 **BLOB (Binary Large Object)**，即较大的二进制对象字段，请看如下程序，注意测试图片自己随便找一个，地址要正确：
首先，在数据库中创建一个表，用于存放图片：

```
CREATE TABLE Images(Id INT PRIMARY KEY AUTO_INCREMENT, Data MEDIUMBLOB);
```

然后运行如下 **PYTHON** 代码进行：

```
# -*- coding: UTF-8 -*-

# 来源：疯狂的蚂蚁的博客 www.crazyant.net 总结整理

import MySQLdb as mdb
import sys

try:
    #用读文件模式打开图片
    fin = open("../web.jpg")
    #将文本读入 img 对象中
    img = fin.read()
    #关闭文件
```

```
fin.close()

except IOError, e:
    #如果出错，打印错误信息
    print "Error %d: %s" % (e.args[0],e.args[1])
    sys.exit(1)

try:
    #链接 mysql，获取对象
    conn = mdb.connect(host='localhost',user='root',passwd='root',
db='test')
    #获取执行 cursor
    cursor = conn.cursor()
    #直接将数据作为字符串，插入数据库
    cursor.execute("INSERT INTO Images SET Data='%s'" %
mdb.escape_string(img))

    #提交数据
    conn.commit()

    #提交之后，再关闭 cursor 和链接
    cursor.close()
    conn.close()

except mdb.Error, e:
    #若出现异常，打印信息
    print "Error %d: %s" % (e.args[0],e.args[1])
    sys.exit(1)
```

结果：



SELECT *
FROM images
LIMIT 0, 30

www.crazyant.net

显示： 起始行： 0 行数： 30 每 100 行重复表头

选项
← T →

	Id	Data
	1	[BLOB - 53 字节]

- `escape_string` 函数将插入数据库的字符串进行转义，这会指一些 SQL 注入的攻击

实例 8、从数据库中把图片读出来

```
# -*- coding: UTF-8 -*-

# 来源：疯狂的蚂蚁的博客 www.crazyant.net 总结整理

import MySQLdb as mdb
import sys

try:
    #连接 mysql，获取连接的对象
    conn = mdb.connect('localhost', 'root', 'root', 'test');

    cursor = conn.cursor()

    #执行查询该图片字段的 SQL
    cursor.execute("SELECT Data FROM Images LIMIT 1")

    #使用二进制写文件的方法，打开一个图片文件，若不存在则自动创建
    fout = open('image.png','wb')
    #直接将数据如文件
    fout.write(cursor.fetchone()[0])
    #关闭写入的文件
    fout.close()

    #释放查询数据的资源
    cursor.close()
    conn.close()

except IOError, e:
    #捕获 IO 的异常，主要是文件写入会发生错误
    print "Error %d: %s" % (e.args[0],e.args[1])
    sys.exit(1)
```

实例 9、使用 Transaction 即事务（手动提交，自动回滚）

```
# -*- coding: UTF-8 -*-

# 来源：疯狂的蚂蚁的博客 www.crazyant.net 总结整理

import MySQLdb as mdb
import sys
```

```
try:
    #连接 mysql, 获取连接的对象
    conn = mdb.connect('localhost', 'root', 'root', 'test');

    cursor = conn.cursor()
    #如果某个数据库支持事务, 会自动开启
    #这里用的是 MySQL, 所以会自动开启事务 (若是 MYISM 引擎则不会)
    cursor.execute("UPDATE Writers SET Name = %s WHERE Id = %s",
        ("Leo Tolstoy", "1"))
    cursor.execute("UPDATE Writers SET Name = %s WHERE Id = %s",
        ("Boris Pasternak", "2"))
    cursor.execute("UPDATE Writer SET Name = %s WHERE Id = %s",
        ("Leonid Leonov", "3"))

    #事务的特性 1、原子性的手动提交
    conn.commit()

    cursor.close()
    conn.close()

except mdb.Error, e:
    #如果出现了错误, 那么可以回滚, 就是上面的三条语句要么执行, 要么都不执行
    conn.rollback()
    print "Error %d: %s" % (e.args[0], e.args[1])
```

结果:

1、因为不存在 writer 表 (SQL 第三条语句), 所以出现错误:

Error 1146: Table 'test.writer' doesn't exist

2、出现错误, 出发异常处理, 3 条语句的前两条会自动变成了没有执行, 结果不变

3、如果本代码放到一个 MyISAM 引擎表, 前两句会执行, 第三句不会; 如果是 INNDB 引擎, 则都不会执行。

全文完, 本文翻译自: <http://zetcode.com/databases/mysqlpythontutorial/>